Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Linear Algebra and its Applications 422 (2007) 616–628
LINEAR ALGEBRA AND ITS APPLICATIONS
www.elsevier.com/locate/laa

A fast and accurate algorithm for solving Bernstein–Vandermonde linear systems

Ana Marco, José-Javier Martínez *

Departamento de Matemáticas, Universidad de Alcalá, Campus Universitario, 28871 Alcalá de Henares, Madrid, Spain

Received 6 May 2006; accepted 14 November 2006
Available online 17 January 2007
Submitted by N.J. Higham

Abstract

A fast and accurate algorithm for solving a Bernstein–Vandermonde linear system is presented. The algorithm is derived by using results related to the bidiagonal decomposition of the inverse of a totally positive matrix by means of Neville elimination. The use of explicit expressions for the determinants involved in the process serves to make the algorithm both fast and accurate.

Keywords: Bernstein basis; Interpolation; Vandermonde matrix; Bidiagonal decomposition; Total positivity; High relative accuracy

1. Introduction

The *Bernstein basis* for the space of algebraic polynomials of degree less than or equal to n is a widely used basis in Computer Aided Geometric Design due to the good properties that it possesses (see, for instance, [5,9–11,17]). However, the explicit conversion between the Bernstein and the power basis is exponentially ill-conditioned as the polynomial degree increases [10]. For this reason, it is very important that when designing algorithms for performing numerical computations with polynomials expressed in Bernstein form, all the intermediate operations are developed using this form only [2]. A paper which presents various basic operations for polynomials in Bernstein

* Corresponding author.
E-mail addresses: ana.marco@uah.es (A. Marco), jjavier.martinez@uah.es (J. Martínez).

doi:10.1016/j.laa.2006.11.020

Table 1
Condition number of a Bernstein–Vandermonde matrix with $x_i = \frac{i}{n+2}$

n	10	15	20	25	30	35	40
$\kappa_\infty(A_n)$	2.1e+04	2.6e+06	3.5e+08	4.7e+10	6.6e+12	9.0e+14	1.3e+17

form is [11]. In [2], an algorithm for computing the greatest common divisor of two polynomials in Bernstein form avoiding explicit basis conversion is given.

Our aim in this paper is to develop a fast and accurate algorithm for solving a linear system whose coefficient matrix is a Bernstein–Vandermonde matrix, that is to say, a generalized Vandermonde matrix for the Bernstein polynomials. Taking into account that this matrix is the coefficient matrix of the linear system associated with a Lagrange interpolation problem in the Bernstein basis, this result will allow us to perform a basic polynomial procedure, the Lagrange interpolation [7], avoiding transformations between Bernstein and power basis. In this way, general algorithms for polynomials in Bernstein form involving an interpolation stage could be designed using this form only.

Our algorithm will be based on the bidiagonal factorization of the inverse of the Bernstein–Vandermonde matrix. Factorizations in terms of bidiagonal matrices are very useful when working with Vandermonde [3,14,16], Cauchy [4], Cauchy–Vandermonde [20,21] and generalized Vandermonde matrices [8].

Let us observe here that, of course, Bernstein–Vandermonde linear systems can be solved by using standard algorithms such as Gaussian or Neville elimination. However they are not fast and the solution provided by them will generally be less accurate since Bernstein–Vandermonde matrices are ill conditioned (see Table 1, where n is the degree of the Bernstein polynomials involved in the definition of the matrix, whose order is $n + 1$) and these algorithms (which do not take into account the structure of the matrix) can suffer from subtractive cancellation.

On the other hand, it must be observed that the large condition numbers are due to the large norm of the inverse matrix since, as it will be easily seen after the definitions of Section 3, the ∞-norm of a Bernstein–Vandermonde matrix is always 1. Therefore, taking into account that if $Ax = b$ and $A(x + \Delta x) = b + \Delta b$ then $\Delta x = A^{-1}\Delta b$, the fact that $\|A^{-1}\|_\infty$ is large implies that the effect of perturbations in the vector b is likely to be important. A lot information about the related concepts of perturbation theory and numerical stability of algorithms in the context of solving linear systems can be found in Chapter 7 of [16].

The rest of the paper is organized as follows. Neville elimination and total positivity are considered in Section 2. In Section 3, the bidiagonal factorization of the inverse of a Bernstein–Vandermonde matrix is presented. In Section 4, the algorithm for solving a linear system whose coefficient matrix is Bernstein–Vandermonde, and the computation of its complexity are given. Finally, Section 5 is devoted to illustrate the accuracy of the algorithm by means of some numerical experiments.

2. Basic results on Neville elimination and total positivity

In this section we will briefly recall some basic results on Neville elimination and total positivity which we will apply in Section 3. Our notation follows the notation used in [12,13]. Given k, $n \in \mathbf{N}$ $(1 \leqslant k \leqslant n)$, $Q_{k,n}$ will denote the set of all increasing sequences of k positive integers less than or equal to n.

Let A be a real square matrix of order n. For $k \leqslant n, m \leqslant n$, and for any $\alpha \in Q_{k,n}$ and $\beta \in Q_{m,n}$, we will denote by $A[\alpha|\beta]$ the submatrix $k \times m$ of A containing the rows numbered by α and the columns numbered by β.

The fundamental tool for obtaining the results presented in this paper is the *Neville elimination* [12,13], a procedure that makes zeros in a matrix adding to a given row an appropriate multiple of the previous one. For a nonsingular matrix $A = (a_{i,j})_{1 \leqslant i,j \leqslant n}$, it consists on $n-1$ steps resulting in a sequence of matrices $A := A_1 \to A_2 \to \cdots \to A_n$, where $A_t = (a_{i,j}^{(t)})_{1 \leqslant i,j \leqslant n}$ has zeros below its main diagonal in the $t-1$ first columns. The matrix A_{t+1} is obtained from A_t $(t = 1, \ldots, n)$ by using the following formula:

$$a_{i,j}^{(t+1)} := \begin{cases} a_{i,j}^{(t)}, & \text{if} \quad i \leqslant t, \\ a_{i,j}^{(t)} - (a_{i,t}^{(t)}/a_{i-1,t}^{(t)})a_{i-1,j}^{(t)}, & \text{if} \quad i \geqslant t+1 \text{ and } j \geqslant t+1, \\ 0, & \text{otherwise}. \end{cases} \tag{2.1}$$

In this process the element

$$p_{i,j} := a_{i,j}^{(j)}, \quad 1 \leqslant j \leqslant n; \quad j \leqslant i \leqslant n$$

is called *pivot* (i, j) of the Neville elimination of A. The process would break down if any of the pivots $p_{i,j}$ $(j \leqslant i < n)$ is zero. In that case we can move the corresponding rows to the bottom and proceed with the new matrix, as described in [12]. The Neville elimination can be done without row exchanges if all the pivots are nonzero, as it will happen in our situation. The pivots $p_{i,i}$ are called *diagonal pivots*. If all the pivots $p_{i,j}$ are nonzero, then $p_{i,1} = a_{i,1}$ $\forall i$ and, by Lemma 2.6 of [12]

$$p_{i,j} = \frac{\det A[i-j+1, \ldots, i|1, \ldots, j]}{\det A[i-j+1, \ldots, i-1|1, \ldots, j-1]}, \quad 1 < j \leqslant i \leqslant n. \tag{2.2}$$

The element

$$m_{i,j} = \frac{p_{i,j}}{p_{i-1,j}}, \quad 1 \leqslant j \leqslant n; \quad j < i \leqslant n \tag{2.3}$$

is called *multiplier* of the Neville elimination of A. The matrix $U := A_n$ is upper triangular and has the diagonal pivots in its main diagonal.

The *complete Neville elimination* of a matrix A consists on performing the Neville elimination of A for obtaining U and then continue with the Neville elimination of U^{T}. The pivot (respectively, multiplier) (i, j) of the complete Neville elimination of A is the pivot (respectively, multiplier) (j, i) of the Neville elimination of U^{T}, if $j \geqslant i$. When no row exchanges are needed in the Neville elimination of A and U^{T}, we say that the complete Neville elimination of A can be done without row and column exchanges, and in this case the multipliers of the complete Neville elimination of A are the multipliers of the Neville elimination of A if $i \geqslant j$ and the multipliers of the Neville elimination of A^{T} if $j \geqslant i$.

A matrix is called *totally positive* (respectively, *strictly totally positive*) if all its minors are non-negative (respectively, positive). The Neville elimination characterizes the strictly totally positive matrices as follows [12]:

Theorem 2.1. *A matrix is strictly totally positive if and only if its complete Neville elimination can be performed without row and column exchanges, the multipliers of the Neville elimination of A and A^{T} are positive, and the diagonal pivots of the Neville elimination of A are positive.*

It is well known [5] that the Bernstein–Vandermonde matrix is a strictly totally positive matrix when the interpolation points satisfy $0 < x_1 < x_2 < \cdots < x_{n+1} < 1$, and this fact has inspired our search for a fast algorithm, but this result will also be shown to be a consequence of our Theorem 3.3.

3. Bidiagonal factorization

The *Bernstein basis* of the space $\Pi_n(x)$ of polynomials of degree less than or equal to n on the interval [0, 1] is:

$$\mathscr{B}_n = \left\{ b_i^{(n)}(x) = \binom{n}{i} (1-x)^{n-i} x^i,\ i = 0, \ldots, n \right\}.$$

We will call the following generalization of the Vandermonde matrices for the Bernstein basis $\mathscr{B}_n$,

$$A = \begin{pmatrix} \binom{n}{0}(1-x_1)^n & \binom{n}{1} x_1(1-x_1)^{n-1} & \cdots & \binom{n}{n} x_1^n \\ \binom{n}{0}(1-x_2)^n & \binom{n}{1} x_2(1-x_2)^{n-1} & \cdots & \binom{n}{n} x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ \binom{n}{0}(1-x_{n+1})^n & \binom{n}{1} x_{n+1}(1-x_{n+1})^{n-1} & \cdots & \binom{n}{n} x_{n+1}^n \end{pmatrix},$$

Bernstein–Vandermonde matrices. Let us observe that these matrices are not of the class of the Vandermonde-like matrices considered in [16], since all the polynomials in the Bernstein basis $\mathscr{B}_n$ have the same degree n and they do not satisfy a three-term recurrence relation. From now on, we will assume $0 < x_1 < x_2 < \cdots < x_{n+1} < 1$.

This matrix A is the coefficient matrix of the linear system associated with the following Lagrange interpolation problem in the Bernstein basis $\mathscr{B}_n$: given the interpolation nodes $\{x_i : i = 1, \ldots, n+1\}$ with $0 < x_1 < x_2 < \cdots < x_{n+1} < 1$ and the interpolation data $\{b_i : i = 1, \ldots, n+1\}$ find the polynomial

$$p(x) = \sum_{k=0}^{n} a_k \binom{n}{k} (1-x)^{n-k} x^k,$$

such that $p(x_i) = b_i$ for $i = 1, \ldots, n+1$. A good introduction to the interpolation theory can be seen in [7].

Preposition 3.1. *For the Bernstein–Vandermonde matrix A defined above we have*:

$$\det A = \binom{n}{0}\binom{n}{1}\cdots\binom{n}{n} \prod_{1 \leqslant i < j \leqslant n+1} (x_j - x_i).$$

Proof. It is easy to see that the matrix of change of basis from the Bernstein basis $\mathscr{B}_n$ to the power basis $\{1, x, x^2, \ldots, x^n\}$ is a lower triangular matrix of order $n+1$ whose diagonal elements are $\binom{n}{0}, \binom{n}{1}, \ldots, \binom{n}{n}$.

From this fact, it is obtained that

$$\det A = \binom{n}{0}\binom{n}{1}\cdots\binom{n}{n}\det V,$$

where V is the Vandermonde matrix

$$V = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n+1} & x_{n+1}^2 & \cdots & x_{n+1}^n \end{pmatrix}.$$

Using the well-known formula for the determinant of a Vandermonde matrix

$$\det V = \prod_{1 \leqslant i < j \leqslant n+1} (x_j - x_i),$$

the proof is concluded. □

The next corollary follows directly from Proposition 3.1, and will be useful to make the derivation of the algorithm easier.

Corollary 3.2

$$\det \begin{pmatrix} (1-x_1)^n & x_1(1-x_1)^{n-1} & \cdots & x_1^n \\ (1-x_2)^n & x_2(1-x_2)^{n-1} & \cdots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ (1-x_{n+1})^n & x_{n+1}(1-x_{n+1})^{n-1} & \cdots & x_{n+1}^n \end{pmatrix} = \prod_{1 \leqslant i < j \leqslant n+1} (x_j - x_i).$$

The following result will be the key to construct our algorithm.

Theorem 3.3. *Let $A = (a_{i,j})_{1 \leqslant i,j \leqslant n+1}$ be a Bernstein–Vandermonde matrix whose nodes satisfy $0 < x_1 < x_2 < \cdots < x_n < x_{n+1} < 1$. Then A^{-1} admits a factorization in the form*

$$A^{-1} = G_1 G_2 \cdots G_n D^{-1} F_n F_{n-1} \cdots F_1, \tag{3.1}$$

where G_i are upper triangular bidiagonal matrices, F_i are lower triangular bidiagonal matrices $(i = 1, \ldots, n)$, and D is a diagonal matrix.

Proof. The matrix A is strictly totally positive (see [5]) and therefore, by Theorem 2.1, the complete Neville elimination of A can be performed without row and column exchanges providing the following factorization of A^{-1} (see [12,13]):

$$A^{-1} = G_1 G_2 \cdots G_n D^{-1} F_n F_{n-1} \cdots F_1,$$

where F_i $(1 \leqslant i \leqslant n)$ are bidiagonal matrices of the form

$$F_i = \begin{pmatrix} 1 & & & & & & & \\ 0 & 1 & & & & & & \\ & \ddots & \ddots & & & & & \\ & & 0 & 1 & & & & \\ & & & -m_{i+1,i} & 1 & & & \\ & & & & -m_{i+2,i} & 1 & & \\ & & & & & \ddots & \ddots & \\ & & & & & & -m_{n+1,i} & 1 \end{pmatrix}, \tag{3.2}$$

G_i^{T} $(1 \leqslant i \leqslant n)$ are bidiagonal matrices of the form

$$G_i^{\mathrm{T}} = \begin{pmatrix} 1 & & & & & & & \\ 0 & 1 & & & & & & \\ & \ddots & \ddots & & & & & \\ & & 0 & 1 & & & & \\ & & & -\tilde{m}_{i+1,i} & 1 & & & \\ & & & & -\tilde{m}_{i+2,i} & 1 & & \\ & & & & & \ddots & \ddots & \\ & & & & & & -\tilde{m}_{n+1,i} & 1 \end{pmatrix}, \tag{3.3}$$

and D is the diagonal matrix whose ith $(1 \leqslant i \leqslant n+1)$ diagonal entry is the diagonal pivot $p_{i,i} = a_{i,i}^{(i)}$ of the Neville elimination of A:

$$D = \operatorname{diag}\{p_{1,1}, p_{2,2}, \ldots, p_{n+1,n+1}\}. \tag{3.4}$$

Taking into account that the minors of A with j initial consecutive columns and j consecutive rows starting with row i are

$$\det A[i, \ldots, i+j-1|1, \ldots, j] = \binom{n}{0}\binom{n}{1}\cdots\binom{n}{j-1}$$
$$(1-x_i)^{n-j+1}(1-x_{i+1})^{n-j+1}\cdots(1-x_{i+j-1})^{n-j+1} \prod_{i \leqslant k<l \leqslant i+j-1}(x_l - x_k),$$

a result that follows from the properties of the determinants and Corollary 3.2, and that $m_{i,j}$ are the multipliers of the Neville elimination of A, we obtain that

$$m_{i,j} = \frac{p_{i,j}}{p_{i-1,j}} = \frac{(1-x_i)^{n-j+1}(1-x_{i-j})\prod_{k=1}^{j-1}(x_i - x_{i-k})}{(1-x_{i-1})^{n-j+2}\prod_{k=2}^{j}(x_{i-1} - x_{i-k})}, \tag{3.5}$$

where $j = 1, \ldots, n$ and $i = j+1, \ldots, n+1$.

As for the minors of A^{T} with j initial consecutive columns and j consecutive rows starting with row i, they are:

$$\det A^{\mathrm{T}}[i, \ldots, i+j-1|1, \ldots, j] = \binom{n}{i-1}\binom{n}{i}\cdots\binom{n}{i+j-2} x_1^{i-1}x_2^{i-1}\cdots x_j^{i-1}$$
$$(1-x_1)^{n-i-j+2}(1-x_2)^{n-i-j+2}\cdots(1-x_j)^{n-i-j+2} \prod_{1 \leqslant k<l \leqslant j}(x_l - x_k).$$

This expression also follows from the properties of the determinants and Corollary 3.2. Since the entries $\tilde{m}_{i,j}$ are the multipliers of the Neville elimination of A^{T}, using the previous expression for the minors of A^{T} with initial consecutive columns and consecutive rows, it is obtained that

$$\tilde{m}_{i,j} = \frac{(n-i+2)\cdot x_j}{(i-1)(1-x_j)}, \quad j=1,\dots,n; \;\; i=j+1,\dots,n+1. \tag{3.6}$$

Finally, the ith diagonal element of D

$$p_{i,i} = \frac{\binom{n}{i-1}(1-x_i)^{n-i+1}\prod_{k<i}(x_i-x_k)}{\prod_{k=1}^{i-1}(1-x_k)}, \quad i=1,\dots,n+1 \tag{3.7}$$

is obtained by using the expression for the minors of A with initial consecutive columns and initial consecutive rows. $\square$

Moreover, by using the same arguments of [20], it can be seen that this factorization is unique among factorizations of this type, that is to say, factorizations in which the matrices involved have the properties shown by formulae (3.2)–(3.4).

Let us observe that the formulae obtained in the proof of Theorem 3.3 for the minors of A with j initial consecutive columns and j consecutive rows, and for the minors of A^{T} with j initial consecutive columns and j consecutive rows show that they are not zero and so, the complete Neville elimination of A can be performed without row and column exchanges. Looking at equations (3.5)–(3.7) is easily seen that $m_{i,j}$, $\tilde{m}_{i,j}$ and $p_{i,i}$ are positive. Therefore, taking into account Theorem 2.1, this confirms that the matrix A is strictly totally positive.

4. The algorithm

In this section we will present a fast algorithm for solving a linear system whose coefficient matrix is a Bernstein–Vandermonde matrix. In order to solve the linear system $Ax = b$, where A is the $(n+1)\times(n+1)$ Bernstein–Vandermonde matrix introduced in Section 3, we use Theorem 3.3 for obtaining

$$x = A^{-1}b = G_1G_2\cdots G_nD^{-1}F_nF_{n-1}\cdots F_1b.$$

Since F_i and G_i $(i=1,\dots,n+1)$ are bidiagonal matrices and D^{-1} is a diagonal matrix, it is clear that the computational complexity of computing the whole product from right to left is $\mathrm{O}(n^2)$. It remains to see that the construction of the matrices F_i, G_i and D^{-1} can be carried out with a computational complexity of $\mathrm{O}(n^2)$.

Let us start with the entries $m_{i,j}$ given by Eq. (3.5). We will use the following expressions:

$$M_{i,1} = \frac{(1-x_i)^n}{(1-x_{i-1})^{n+1}},$$
$$m_{i,1} = (1-x_{i-1})\cdot M_{i,1},$$
$$M_{i,j} = \frac{(1-x_i)^{n-j+1}\prod_{k=1}^{j-1}(x_i-x_{i-k})}{(1-x_{i-1})^{n-j+2}\prod_{k=2}^{j}(x_{i-1}-x_{i-k})},$$
$$M_{i,j+1} = \frac{(1-x_{i-1})(x_i-x_{i-j})}{(1-x_i)(x_{i-1}-x_{i-j-1})}\cdot M_{i,j},$$
$$m_{i,j+1} = (1-x_{i-j-1})\cdot M_{i,j+1},$$

where $i=2,\dots,n+1$ and $j=1,\dots,i-2$, in their construction:

for $i = 2 : n + 1$
  $M = \frac{(1-x_i)^n}{(1-x_{i-1})^{n+1}}$
  $m_{i,1} = (1 - x_{i-1}) \cdot M$
  for $j = 1 : i - 2$
    $M = \frac{(1-x_{i-1})(x_i - x_{i-j})}{(1-x_i)(x_{i-1} - x_{i-j-1})} \cdot M$
    $m_{i,j+1} = (1 - x_{i-j-1}) \cdot M$
  end
end

Now we compute the entries $\tilde{m}_{i,j}$ given by Eq. (3.6):

for $j = 1 : n$
  $c_j = \frac{x_j}{1-x_j}$
  for $i = j + 1 : n + 1$
    $\tilde{m}_{i,j} = \frac{n-i+2}{i-1} \cdot c_j$
  end
end

As for the diagonal elements $p_{i,i}$ of D given by Eq. (3.7), they are constructed using the equalities

$$
\begin{aligned}
q_{1,1} &= 1, \\
p_{1,1} &= (1 - x_1)^n, \\
q_{i,i} &= \frac{\binom{n}{i-1}}{\prod_{k=1}^{i-1}(1 - x_k)}, \\
q_{i+1,i+1} &= \frac{n - i + 1}{i \cdot (1 - x_i)} \cdot q_{i,i}, \\
p_{i+1,i+1} &= q_{i+1,i+1} \cdot (1 - x_{i+1})^{n-i} \prod_{k<i+1} (x_{i+1} - x_k),
\end{aligned}
$$

where $i = 1, \ldots, n$, in the following way:

$q = 1$
$p_{1,1} = (1 - x_1)^n$
for $i = 1 : n$
  $q = \frac{(n-i+1)}{i(1-x_i)} \cdot q$
  $aux = 1$
  for $k = 1 : i$
    $aux = (x_{i+1} - x_k) \cdot aux$
  end
  $p_{i+1,i+1} = q \cdot (1 - x_{i+1})^{n-i} \cdot aux$
end

Looking at this algorithm is enough to conclude that the computational complexity of the construction of the matrices D, F_i and G_i $(i = 1, \ldots, n + 1)$ is $O(n^2)$, and therefore, the computational complexity of solving the whole linear system is also $O(n^2)$.

A similar algorithm with computational complexity $O(n^2)$ can be developed for computing the bidiagonal factorization of the inverse of A^T, that is, A^{-T}. In consequence, an algorithm with computational complexity $O(n^2)$ is obtained for solving the dual linear system $A^{\mathrm{T}}x = b$, where A is an $(n+1) \times (n+1)$ Bernstein–Vandermonde matrix, by using:

$$x = A^{-\mathrm{T}}b = F_1^{\mathrm{T}} F_2^{\mathrm{T}} \cdots F_n^{\mathrm{T}} D^{-1} G_n^{\mathrm{T}} G_{n-1}^{\mathrm{T}} \cdots G_1^{\mathrm{T}}.$$

5. Numerical experiments and final remarks

Finally we present some numerical experiments which illustrate the good properties of our algorithm. We compute the exact solution x_e of each one of the Bernstein–Vandermonde linear systems $Ax = b$ by using the command `linsolve` of *Maple 10* and use it for comparing the accuracy of the results obtained in MATLAB by means of:

(1) The algorithm presented in Section 4 for computing the bidiagonal decomposition of A^{-1}. We will call it `MM`.
(2) The algorithm `TNBD` of Plamen Koev [18] that computes the bidiagonal decomposition of A^{-1} without taking into account the structure of A.
(3) The command $A\backslash b$ of MATLAB.

The fast product (from right to left) of the bidiagonal matrices and the vector b is also implemented in MATLAB and is the second stage in the solution of the linear system in (1) and (2).

We compute the relative error of a solution x of the linear system $Ax = b$ by means of the formula:

$$\mathrm{err} = \frac{\|x - x_e\|_2}{\|x_e\|_2}.$$

Remark. The algorithm `TNBD` computes the matrix denoted as $\mathscr{BD}(A)$ in [19], which represents the *bidiagonal decomposition* of A. But it is a remarkable fact that the same matrix $\mathscr{BD}(A)$ also serves to represent the bidiagonal decomposition of A^{-1}. The algorithm computes $\mathscr{BD}(A)$ by performing *Neville elimination* on A, which involves true substractions, and therefore does not guarantee high relative accuracy.

A detailed error analysis of Neville elimination, which shows the advantages of this type of elimination for the class of totally positive matrices, has been carried out in [1], and related work for the case of Vandermonde linear systems can be seen in Chapter 22 of [16].

Example 5.1. Let $\mathscr{B}_{10}$ be the Bernstein basis of the space of polynomials with degree less than or equal to 10 on [0, 1] and A be the Bernstein–Vandermonde matrix of order 11 generated by the nodes $x_i = \frac{i}{12}$ for $i = 1, \ldots, 11$. The condition number of A is $\kappa_2(A) = 1.8e+04$. Let us consider

$$b_1^{\mathrm{T}} = (1, 0, 2, -1, 3, 1, -2, 0, 0, 3, 5)^{\mathrm{T}},$$
$$b_2^{\mathrm{T}} = (1, -2, 1, -1, 3, -1, 2, -1, 4, -1, 1)^{\mathrm{T}},$$

two vectors of data.

The relative errors obtained when using the approaches (1), (2) and (3) for solving the systems $Ax = b_i$ $(i = 1, 2)$ are reported in Table 2.

Table 2
Example 5.1: A is a Bernstein–Vandermonde matrix of order 11

b_i	*MM*	*TNBD*	$A\backslash b_i$
b_1	1.3e−15	7.8e−14	5.4e−14
b_2	8.6e−16	8.2e−14	1.0e−14

The following example will show how the accuracy of our approach is maintained when the order, and therefore the condition number (see Table 1), of the Bernstein–Vandermonde matrix increases, while the accuracy of the other two approaches which do not exploit the structure of the matrix A goes down.

Example 5.2. Let $\mathscr{B}_{15}$ be the Bernstein basis of the space of polynomials with degree less than or equal to 15 on [0, 1] and A be the Bernstein–Vandermonde matrix of order 16 generated by the nodes $x_i = \frac{i}{17}$ for $i = 1, \dots, 16$. The condition number of A is $\kappa_2(A) = 2.3e + 06$. Let

$$b_1^{\mathrm{T}} = (2, 1, 2, 3, -1, 0, 1, -2, 4, 1, 1, -3, 0, -1, -1, 2)^{\mathrm{T}},$$
$$b_2^{\mathrm{T}} = (1, -2, 1, -1, 3, -1, 2, -1, 4, -1, 2, -1, 1, -3, 1, -4)^{\mathrm{T}},$$

two vectors containing the data. The relative errors of the solutions of the linear systems $Ax = b_i$ $(i = 1, 2)$ obtained by means of the approaches (1)–(3) are reported in Table 3.

Let us observe that in the first stage of the algorithm, which corresponds to the computation of the bidiagonal decomposition of A^{-1}, the *high relative accuracy* of the algorithm is obtained because no subtractive cancellation occurs: we are multiplying, dividing, or adding quantities with the same sign, or forming $1 - x_i$ and $x_i \pm x_j$ where x_i and x_j are initial data.

Remark. Although in this paper we are only considering the problem of solving Bernstein–Vandermonde linear systems, it must be observed that the first stage of our algorithm can also be used as an intermediate step for the computation of the eigenvalues and the singular value decomposition of a Bernstein–Vandermonde matrix since, as it is shown in [19], *given the bidiagonal factors of a nonsingular totally nonnegative matrix its eigenvalues and its SVD can be computed to high relative accuracy in floating point arithmetic*, *independent of the conventional condition number*.

As for the second stage of the algorithm corresponding to the evaluation of the product

$$G_1 G_2 \cdots G_n D^{-1} F_n F_{n-1} \cdots F_1 b,$$

it must be noted that when the vector b has alternating sign pattern $(\mathrm{sign}(b_i) = (\pm 1)^i)$, this stage is substraction-free, and so the product can be computed with high relative accuracy. This is a

Table 3
Example 5.2: A is a Bernstein–Vandermonde matrix of order 16

b_i	*MM*	*TNBD*	$A\backslash b_i$
b_1	1.0e−15	5.9e−11	6.5e−12
b_2	4.9e−16	5.9e−11	6.4e−12

consequence to the checkerboard sign pattern of A^{-1}, which derives from the fact that A is a totally positive matrix.

This important property was already observed in an analogous situation in the paper [15], devoted to the error analysis of the Björck–Pereyra algorithm for Vandermonde systems. There the use of the increasing ordering for the interpolation points is recommended, an ordering which in the case $0 < x_1 < x_2 < \cdots < x_n$ makes the Vandermonde matrix a totally positive one. More recently this fact has also been pointed out in [8], where the Björck–Pereyra algorithm for ordinary Vandermonde matrices is extended to the case of totally positive generalized Vandermonde linear systems.

The results of Example 5.1 (shown in Table 2) and Example 5.2 (shown in Table 3) illustrate this fact: in both cases b_2 has alternating sign pattern and consequently the relative accuracy of our algorithm is higher for b_2 than for b_1.

On the other hand, in those examples even for b_1 the results obtained by our algorithm are better than the results obtained when using the other two algorithms. Moreover, the advantage of our algorithm is greater when the order of the Bernstein–Vandermonde matrix increases.

Our next two examples will serve to illustrate the concept of *effective well-conditioning* introduced by Chan and Foulser in [6]. This concept has also been studied in the context of totally positive Cauchy systems in [4], where it is seen that the *BKO* algorithm exploits the effective well-conditioning to produce higher accuracy for special right-hand sides.

Let $A \in R^{n\times n}$ with singular value decomposition $A = U\Sigma V^{\mathrm{T}}$, where $\Sigma = \mathrm{diag}(\sigma_i)$, $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n$ and $U = [u_1 u_2 \cdots u_n]$. Let $P_k = U_k U_k^{\mathrm{T}}$, with $U_k = [u_{n+1-k} \cdots u_n]$, the projection operator onto the linear span of the smallest k left singular vectors of A. The *Chan–Foulser number* [4,6] for the linear system $Ax = f$ is defined by

$$\gamma(A, f) = \min_k \frac{\sigma_{n-k+1}}{\sigma_n} \frac{\|f\|_2}{\|P_k f\|_2}.$$

Let us observe that the computation of the Chan–Foulser number simplifies when $f = u_i$ ($i = 1, \ldots, n$). In this case, using the fact that the left singular vectors $u_1, \ldots, u_n$ are the columns of the orthogonal matrix U, we obtain

$$\gamma(A, u_i) = \frac{\sigma_i}{\sigma_n}.$$

Example 5.3. Let A be the same matrix as the one considered in Example 5.2, whose condition number is $\kappa_2(A) = 2.3e + 06$. We have solved the 16 linear systems $Ax = u_i$, where $u_1, \ldots, u_{16}$ are the left singular vectors of A. The results obtained when using the approaches (1)–(3), and the corresponding Chan–Foulser numbers are reported in Table 4.

Example 5.4. Let $\mathscr{B}_{15}$ be the Bernstein basis of the space of polynomials with degree less than or equal to 15 on [0, 1] and A be the Bernstein–Vandermonde matrix of order 16 generated by the interpolation nodes

$$\frac{1}{18} < \frac{1}{16} < \frac{1}{14} < \frac{1}{12} < \frac{1}{10} < \frac{1}{8} < \frac{1}{6} < \frac{1}{4} < \frac{11}{20} < \frac{19}{34} < \frac{17}{30} < \frac{15}{26} < \frac{11}{18} < \frac{9}{14} < \frac{7}{10} < \frac{5}{6}.$$

Its condition number is $\kappa_2(A) = 3.5e + 09$. We have solved the 16 linear systems $Ax = u_i$, where $u_1, \ldots, u_{16}$ are the left singular vectors of A.

The results obtained when using the approaches (1)–(3), and the corresponding Chan–Foulser numbers are reported in Table 5.

The results appearing in Tables 4 and 5 show that, as it could be expected, the solution vectors obtained by conventional methods based on Neville elimination or on Gaussian elimination suffer from a loss of digits of precision which is close to the decimal logarithm of the condition number of the Bernstein–Vandermonde matrix, a condition number which does not depend on the right-hand side vector b.

On the contrary, the results in Tables 4 and 5 indicate that the accuracy obtained by our algorithm, which exploits the structure of the matrix, is governed by the Chan–Foulser number: the relative accuracy increases when the Chan–Foulser condition number decreases. Inparticu-

Table 4
Example 5.3: left singular vectors u_i for the right hand side

u_i	$\gamma(A, u_i)$	*MM*	*TNBD*	$A\backslash u_i$
u_1	2.3e+06	1.1e−10	4.7e−12	2.8e−11
u_2	2.1e+06	5.0e−11	1.0e−11	1.7e−11
u_3	1.7e+06	2.5e−11	1.3e−11	1.6e−10
u_4	1.3e+06	4.9e−11	3.8e−11	3.4e−11
u_5	9.3e+05	4.3e−11	1.1e−11	1.4e−11
u_6	6.0e+05	3.1e−11	1.7e−11	6.2e−12
u_7	3.5e+05	4.0e−11	7.5e−12	2.1e−11
u_8	1.8e+05	1.8e−12	1.8e−11	1.4e−12
u_9	8.5e+04	1.2e−11	9.8e−12	6.0e−12
u_{10}	3.4e+04	1.7e−12	1.5e−11	9.1e−12
u_{11}	1.2e+04	4.9e−13	2.0e−11	3.1e−12
u_{12}	3.4e+03	6.5e−13	1.4e−11	1.4e−11
u_{13}	7.9e+02	1.4e−13	2.4e−11	2.4e−11
u_{14}	1.4e+02	8.1e−14	6.3e−12	2.4e−11
u_{15}	1.6e+01	7.1e−15	2.1e−11	1.3e−11
u_{16}	1	5.1e−16	5.9e−11	6.3e−12

Table 5
Example 5.4: left singular vectors u_i for the right hand side

u_i	$\gamma(A, u_i)$	*MM*	*TNBD*	$A\backslash u_i$
u_1	3.5e+09	3.5e−07	3.2e−08	4.6e−08
u_2	2.6e+09	1.1e−07	2.4e−08	7.3e−08
u_3	1.3e+09	2.9e−08	1.3e−08	3.7e−08
u_4	1.2e+09	2.2e−08	1.2e−08	5.0e−09
u_5	5.3e+08	2.7e−08	2.3e−08	5.5e−09
u_6	4.0e+08	2.5e−09	1.1e−08	4.6e−08
u_7	1.1e+08	3.6e−09	2.4e−08	8.5e−09
u_8	5.8e+07	2.6e−09	3.2e−09	8.0e−09
u_9	1.1e+07	2.4e−10	1.1e−08	2.5e−08
u_{10}	3.7e+06	3.9e−10	5.3e−09	8.1e−09
u_{11}	4.8e+05	4.8e−12	8.0e−09	1.3e−08
u_{12}	1.2e+05	1.5e−11	1.1e−08	2.2e−08
u_{13}	6.2e+03	2.4e−12	1.5e−08	3.3e−08
u_{14}	9.3e+02	7.3e−13	2.8e−10	1.9e−08
u_{15}	1.4e+01	4.3e−14	3.0e−09	3.6e−08
u_{16}	1	7.6e−15	1.1e−09	1.4e−08

lar, the very high relative precision observed in the 16th system of both examples reflects two facts: the Chan–Foulser number is equal to 1 and the vector u_{16} has alternating sign pattern ($\mathrm{sign}(b_i) = (\pm 1)^i$). This fact is also pointed out for the case of Cauchy linear systems in [4], where it is recalled that *the last left singular vector corresponding to the smallest singular value has the sign-interchanging property*.

Acknowledgements

This research has been partially supported by Spanish Research Grant MTM 2006-03388 from the Spanish Ministerio de Ciencia y Tecnología. We are grateful to the referees for their suggestion of extending the numerical experiments.

References

[1] P. Alonso, M. Gasca, J.M. Peña, Backward error analysis of Neville elimination, Appl. Numer. Math. 23 (1997) 193–204.
[2] D.A. Bini, L. Gemignani, Bernstein–Bezoutian matrices, Theor. Comput. Sci. 315 (2004) 319–333.
[3] A. Björck, V. Pereyra, Solution of Vandermonde systems of equations, Math. Comput. 24 (1970) 893–903.
[4] T. Boros, T. Kailath, V. Olshevsky, A fast parallel Björck–Pereyra-type algorithm for solving Cauchy linear equations, Linear Algebra Appl. 302/303 (1999) 265–293.
[5] J.M. Carnicer, J.M. Peña, Shape preserving representations and optimality of the Bernstein basis, Adv. Comput. Math. 1 (1993) 173–196.
[6] T. Chan, D. Foulser, Effectively well-conditioned linear systems, SIAM J. Sci. Stat. Comput. 9 (1998) 963–969.
[7] P.J. Davis, Interpolation and Approximation, Dover Publications, New York, 1975.
[8] J. Demmel, P. Koev, The accurate and efficient solution of a totally positive generalized Vandermonde linear system, SIAM J. Matrix Anal. Appl. 27 (1) (2005) 142–152.
[9] G. Farin, Curves and Surfaces for CAGD: A Practical Guide, fifth ed., Academic Press, San Diego, 2002.
[10] R.T. Farouki, On the stability of transformations between power and Bernstein polynomial forms, Comput. Aided Geom. Design 8 (1991) 29–36.
[11] R.T. Farouki, V.T. Rajan, Algorithms for polynomials in Bernstein form, Comput. Aided Geom. Design 5 (1988) 1–26.
[12] M. Gasca, J.M. Peña, Total positivity and Neville elimination, Linear Algebra Appl. 165 (1992) 25–44.
[13] M. Gasca, J.M. Peña, A matricial description of Neville elimination with applications to total positivity, Linear Algebra Appl. 202 (1994) 33–45.
[14] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, 1996.
[15] N.J. Higham, Error analysis of the Björck–Pereyra algorithms for solving Vandermonde systems, Numer. Math. 50 (5) (1987) 613–632.
[16] N.J. Higham, Accuracy and Stability of Numerical Algorithms, second ed., SIAM, Philadelphia, 2002.
[17] J. Hoschek, D. Lasser, Fundamentals of Computer Aided Geometric Design, A.K. Peters, Wellesley, 1993.
[18] P. Koev, http://www-math.mit.edu/~plamen.
[19] P. Koev, Accurate eigenvalues and SVDs of totally nonnegative matrices, SIAM J. Matrix Anal. Appl. 27 (2005) 1–23.
[20] J.J. Martínez, J.M. Peña, Factorizations of Cauchy–Vandermonde matrices, Linear Algebra Appl. 284 (1998) 229–237.
[21] J.J. Martínez, J.M. Peña, Factorizations of Cauchy-Vandermonde matrices with one multiple pole, in: O. Pordavi (Ed.), Recent Research on Pure and Applied Algebra, Nova Sci., Hauppage, NY, 2003, pp. 85–95.